What Should Markup Really Be? Applying theories of text to the design of markup systems

David G. Durand, Elli Mylonas, Steven J. DeRose

David G. Durand, 12 Harbour Terrace, Cranston, RI, 02905
Elli Mylonas, 12 Harbour Terrace, Cranston, RI, 02905
Steven J. DeRose, Electronic Book Technologies, 1 Richmond square, Providence, RI 02906

KEYWORDS: SGML, text encoding, markup theory

AFFILIATION: Boston University, Brown University, Electronic Book Technologies

E-MAIL:	dgd@cs.bu.edu Elli_Mylonas@brown.edu sjd@ebt.com
FAX NUMBER:	+1 (401)-331-2015
PHONE NUMBER:	+1 (401)-331-2014 (work) +1 (401)-781-5137 (home)

Introduction

The issue of what text really is, and how it affects our notions of proper text representation has been with us almost from the beginning of text encoding [Goldfarb 1981, Reid 1980, Coombs, et al. 1987, DeRose, et al. 1990, Renear et al.]. The simplest reasonable view, that text is fundamentally an ordered hierarchical structure, determined by its editor and author, is an early one that has remained prominent, especially as reified by ISO 8879 (SGML). However, this simple model is not enough, which the TEI [Sperberg-McQueen and Burnard 1990,1993] quickly discovered as it moved text encoding from the realm of print production to that of scholarship, textual editing, and linguistic analysis. The TEI metalanguage committee identified problems with SGML's simple hierarchical mechanisms, and developed and published techniques for working around them to encode non-hierarchical phenomena [Barnard et al. 1996]. In [Renear et al.] we began to analyze and label the theoretical and ontological foundations underlying many of the kinds of non-hierarchical structures discovered by practitioners using naive hierarchical markup. This paper uncovered some key notions and implicit partial theories underlying most previous theorizing about markup. The most important of these notions is the primacy of "analytic perspectives," which we defined as a "natural family of methodology, theory, and analytical practice." Perspectives explain various implicit presuppositions of the simple hierarchical approach. In this paper, we use these theoretical results to examine how the basic notions of hierarchical markup should be extended to allow a more expressive and accurate approach to document markup.

Most of the following discussion is framed in terms of SGML, because SGML represents the state of the art in document description languages. The features that we propose can be regarded either as sugggestions for improving SGML, as specifications for some future successor, or even specifications for a new standard, that, like HyTime, would add additional power to SGML markup. We do not take a position on these thorny standards issues, concentrating rather on the problems to be addressed. In our examples we will use syntax based on SGML for clarity, but we will diverge from that syntax as necessary (and with explanation).

Some Phenomena

The following (partial) list of non-hierarchical phenomena is based on [Renear et al., and Barnard, et al. 1995]:

- Arbitrary overlaps caused by multiple perspectives on a text. For instance, metrical and grammatical structuring of the same poem have no essential relation to each other, and many phenomena of interest within one perspective, like phrases or clauses, will overlap arbitrarily with the phenomena of interest from the other perspectives, like stanzas or verse lines.
- Arbitrary overlaps of entities that are important to differing sub-perspectives of a single perspective on a text.
- Discontinuous content objects, such as interrupted lists, interrupted quotations or speeches.
- "Partial" perspectives, like a metrical analysis of a mixed prose/verse work. For the verse segments of the work, the metrical perspective is valid, while for the prose portions the metrical perspective does not even exist.
- Segmentations of a text, like page number assignment or position in a formal reference system, where every part of a text is supposed to have a certain value for a given attribute.
- Truly independent objects, like hypertext links, which might overlap arbitrarily.
- Ambiguous content, like a sentence that can be parsed in several ways. Such content is part of a single perspective, which needs to record more than one possible analysis of the same item.

Methods of tagging all of these currently exist in the TEI, in the form of particular tags for particular perspectives. But since we now know that the breaking of strict hierarchies is the rule, rather than the exception, it is time to determine what additional features are required from markup systems to make the formal description of such non-hierarchical phenomena straightforward. We propose that it is better to integrate the formal properties of these recurring non-hierarchical phenomena into markup systems themselves, rather than re-inventing them tag-by-tag. Their explicit representation will enable more perspicuous, explicit, and consistent descriptions of nonhierarchical tag-relationships and constraints, in the same way as the formal definitions of content models in SGML do for hierarchical documents.

The feature-structure tags in the TEI [Langendoen 1995] are actually general enough to handle any non-hierarchical structure. However, featurev structures are not appealing because their consistent application to the problem of general document markup would lead to documents containing no tags other than the feature-structure tags, with all the information that is currently represented by tags encoded in them. This would produce extremely verbose encodings that would not take advantage of the tags syntax on which they are based. In short, feature structures do not solve the problem of *extending* hierarchical markup systems to deal with non-hierarchical markup structures; rather, they solve the related problem of encoding non-hierarchical structures *within* a hierarchical markup system.

Representing Non-Hierarchical Features and Relationships

One of the most obvious points to start with is the notion of analytical perspectives. The SGML CONCUR feature comes close to expressing the basic notion, but has a number of serious defects:

- It cannot deal with incomplete hierarchies easily, since the standard requires that each concurrent stream be a complete DTD.
- It does not allow tag content that is meaningful in only a single perspective, by requiring that characters that occur anywhere in the document *must* be visible in all concurrent views.
- It does not support the notion of sub-perspectives, since there is no way to express relations between different concurrent markup structures for a document. This might be used, for instance, to express the fact that a metrical perspective (and its associated markup) is only applicable within text that is tagged as a poem. The SGML LINK feature, which seems superficially to address this problem is too processing-oriented and poorly defined to solve this problem.
- A final problem with CONCUR is that it has only been implemented once as far as the authors know. Certain syntactic irregularities occur in defining the interaction of concurrent markup streams with SGML's minimization features, rendering correct implementation extremely difficult.

The problem of arbitrarily overlapping segments, like hypertext anchors, is usually handled by the use of SGML EMPTY tags and IDREFs. The problem with this appproach is that the DTD cannot indicate the usage of such tags. For instance, given two tags <startSeg> and <endSeg> a DTD cannot indicate:

- that <startSeg> and <endSeg> are intended to be paired
- that it is the text between them that is important when they are processed,
- that they must refer to each other in pairs and not to any other tags
- that a paired <startSeg> and <endSeg> should appear in the text such that the <startSeg> occurs first
- that paired <startSeg> and <endSeg> tags could have some specified relation to other markup in the document, for instance, that they should not not overlap paragraph boundaries

The issue here has nothing to do with the syntax or the use of cross-references to indicate relationships in the document instance. Rather, it is whether the processing system can automatically perform the obvious useful verification tasks, or enforce user requirements that depend on the intended semantics of the tags (such as any relationships to other tags that should be enforced or forbidden).

Ambiguous content, as described in [Barnard, et al. 1995] is content that has several differing analyses within a given perspective. It is especially interesting because the phenomenon that it reflects is so important. The points where analytic ambiguity exists, are often the most interesting ones in many different disciplines. There are other interesting aspects of the markup of ambiguous texts. For instance, a document that records ambiguity precisely, sharing structure down to the lexical level, is different from one that records ambiguous structures at the the sentence level, sharing only the leaf text, despite a large number of identical structures.

Segmentations of a text, like reference systems that assign portions of a text to one of a discrete set of regions, can be analyzed as an extremely simple special case of hierarchical markup: a single level hierarchy with tags that divide an entire document into segments. Since such structures are usually marked when they do not have a natural mapping to a hierarchical perspective, they are candidates for special treatment. This kind of structure is usually marked by empty “milestone” tags.

Some Proposals

Based on the fundamental sorts of non-hierarchical structure described in the discussion above, we will briefly discuss the kinds of additions needed to accomodate each phenomenon.

Multiple perspectives

This is a fairly simple concept, similar to using multiple clear overlays of a page, each marked with highlighter. We propose to represent different perspectives by *streams*. A stream is a set of markup objects corresponding to a perspective in the same way that an SGML element corresponds with a document object. Streams are similar to CONCUR, except that:

- A stream can declare some of its content (elements and character data) *private*, so that it is invisible to other streams.
- A stream can exclude data based on how it is marked in other streams
- There is no requirement that a parser present a single-stream view to a client program. The logical view of a document should include as many streams as the processor requires.

Arbitrary overlaps of sub-perspectives

We treat each sub-perspective as an independent stream, and express the relationships between sub-perspectives by allowing the declaration of constraints on the co-occurrence of elements in the streams.
We add:

- A declaration that an element and its sub-elements will be contained within the boundaries of a particular element type (which might possibly be in another stream). This allows a document designer to explicitly limit the range over which elements might overlap each other so that a certain stream, like one containing metrical information, should only apply in areas marked in accordance with a stream that indicates verse stanzas..

Partial Perspectives

We extend the notion of a stream to have several “root elements.” A list of possible top-level elements in the stream is given, and those may occur anywhere in the document (as long as they meet the other constraints defined for them). All text not contained by any of the roots is automatically ignored in that stream.

Segmentations

Since a segmentation is simply a specialized, but common, form of stream, we allow it as a special type. When declared a segmentation stream can optionally specify an element (of some stream). If specified this state that the segmentation is valid (and required) in every occurence of that element. If no such element is specified, at least one of the elements in the segmentation stream must occur before any character content in the document. All these elements are represented by EMPTY tags and interpreted as elements spanning a region from their appearance, up to the next occurrence of a tag in the same stream.

Independent objects

These cannot be handled properly by the stream notion (as we noted in the discussion of CONCUR). Elements like this may overlap themselves. To handle such elements we need only add a way to specify where they can break the hierarchy within their own stream – any requirements to constrain an element not to break the hierarchy in other streams have already been handled above.
We allow two new hierarchy-breaking declarations:

- A declaration that the element can overlap itself, with an optional number to limit its degree of deepest self-overlap. Declaring this for an annotation element would allow arbitrarily scoped, arbitrarily related zones of a text to be marked without difficulty
- A declaration that the element can overlap with a particular tag or list of tags. For each tag (overlapee) that the element might overlap, there is an optional specification of whether that it is permitted to overlap the children of the overlapee.

Multiple interpretations

These do not have to be specially handled as they are formally similar to self-overlapping tags. All that is required is to declare such elements as self-overlapping within the stream corresponding to their perspective.

Conclusions

The proposed extensions of document schemas to handle non-hierarchical markup represent the results of many years of experience with text-encoding in the humanities community. We feel that they are an important starting point for improving the quality of humanities computing tools for the next generation of markup systems.

Bibliography

Barnard, David, Burnard, Lou, Gaspart, Jean-Pierre, Price, Lynne A., Sperberg-McQueen, C. M., Varile, Giovanni Batista. “Hierarchical Encoding of Text: Technical Problems and SGML Solutions” Computers and the Humanities. 29 (1995): 211-231.

Coombs, James S., Allen H. Renear and Steven J. DeRose. “Markup Systems and the Future of Scholarly Text Processing” Communications of the Association for Computing Machinery. 30 (1987): 933-47.

DeRose, Steven, J., David Durand, Elli Mylonas and Allen H. Renear. “What is Text, Really?,” Journal of Computing in Higher Education. 1:2 (1990).

Goldfarb, Charles. “A Generalized Approach to Document Markup.” in Proceedings of the ACM SIGPLAN–SIGOA Symposium on Text Manipulation, New York: ACM, 1981.

Langendoen, D. Terence, Simons, Gary F., “Rationale for the TEI Recommendations for Feature-Structure Markup” Computers and the Humanities. 29 (1995): 191-209.

Reid, Brian. “A High-Level Approach to Computer Document Formatting.” in Proceedings of the 7th Annual ACM Symposium on Programming Languages. New York: ACM, 1980.

Renear, Allen. David Durand, and Elli Mylonas. “Refining Our Notion of What Text Really Is.” Research in Humanities Computing. Oxford: Oxford University Press, forthcoming.

Sperberg-McQueen, C. Michael. and Burnard, Lou, eds. Guidelines for the Encoding and Interchange of Machine-Readable Texts.. Chicago and Oxford: TEI, 1990, 1993.